REMARKS

Applicant amended independent claim 1 to clarify that the associated bit mask is specified in the executed instruction that causes the selective loading of data into the transfer register. Support for this clarification is provided, for example, at page 11, lines 8-12 of the originally filed application. Independent claims 10 and 19 were similarly amended. After these amendments, claims 1, 3-10, 12-19, and 21-27 are pending in the above-identified patent application. Claims 1, 10 and 19 are independent claims.

The examiner rejected claims 1, 3, 5, 7 and 8 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,487,159 to Byers, in view of the reference "Computer Organization and Design" by Hennessey and Patterson, and further in view of the reference "Computer Architecture, a Quantitative Approach", also by Hennessey and Patterson.

The examiner also rejected claim 4 under 35 U.S.C. §103(a) as being unpatentable over Byers in view of in view of the "Computer Organization and Design" reference, in view of the "Computer Architecture, a Quantitative Approach" reference, and further in view of U.S. Patent No. 4,569,016 to Hao.

The examiner further rejected claim 9 under 35 U.S.C. §103(a) as being unpatentable over Byers in view of in view of the "Computer Organization and Design" reference in view of the "Computer Architecture, a Quantitative Approach" reference, in view of Hao, and further in view of U.S. Patent No. 5,832,258 to Kiuchi.

Additionally, the examiner rejected claims 10, 12, 14, 16, 17, 19, 21, 23, 25, and 26 under 35 U.S.C. §103(a) as being unpatentable over Byers in view of in view of the "Computer Organization and Design" reference.

The examiner rejected claims 13, 15, 22 and 24 under 35 U.S.C. §103(a) as being unpatentable over Byers in view of in view of the "Computer Organization and Design" reference, and further in view of Hao.

The examiner also rejected claims 18 and 27 under 35 U.S.C. §103(a) as being unpatentable over Byers in view of in view of the "Computer Organization and Design" reference, in view of the "Computer Architecture, a Quantitative Approach" reference, in view of Hao, and further in view of Kiuchi.

Applicant's independent claim 1 recites "one of the instructions causing the execution box to selectively load any specified combination of bytes of data within a transfer register associated with one of the plurality of microengines, with a shifted value of an operand that preserves the bytes of data that are not loaded, with loading using an associated bit mask specified in the instruction with each bit of the associated bit mask identifying a different byte of the transfer register." Thus, the mask that is used to facilitate selectively loading data into the transfer register is specified in the instruction that causes the performance of the selective data loading.

Byers describes a digital computer system that performs shift, mask and merge operations on operands in a single instruction (col. 1, lines 20-22). Particularly, as explained by Byers:

> **FIG. 2 is a diagram of the Arithmetic Logic and Shift Internal (ALUI) instruction format. ...**
>
> **When a shift operation is requested (S=1), the L field (bit 6) 56 indicates whether the shift is a left shift (when L=1) or a right shift (when L=0). The Count field (bits 7-10) 58 specifies the number of bits to be shifted. The Source A field (bits 12-21) 60 indicates the address in the Local Store 16 where the first operand for this operation is to be fetched. The first operand is loaded into Register A 22 as pan of the processing of this instruction. The second operand was loaded into the Bus Received Register 20 from the Micro Bus 18 during the execution of the previous instruction. The Mask Source field (bits 22-31) 62 indicates the address in the Local Store 16 where the Mask and Merge operand is to be fetched. The Mask and Merge operand is stored in Mask/Merge Register 24.**
>
> **FIG. 3 shows the format of the Mask and Merge operand. The even numbered bits (denoted by `M`) represent the Mask values and the odd bits (denoted by `R`) represent the Merge values. These bits control the masking and merging of the data. The Mask/Merge bits operate on bit pairs of an operand. Thus, a Mask/Merge bit pair (bits 0-1, 2-3, 4-5, etc.) control corresponding bit pairs of the operand. If a Mask bit is set, then the corresponding bit pair of the operand is carried forward to the result register. If a Mask bit is clear, then the corresponding bit pair of the operand is not carried forward to the result register; instead, the specified bit pair in the result register is cleared. If a Merge bit is set, the corresponding bit pair of the second operand, which is stored in Bus Received Register 20, is copied to the result register. If a Merge bit is clear, the corresponding bit pair of the first operand, which is stored in Register A 22, is copied to the result register.**
>
> ...
>
> **Referring back to FIG. 2, when the R field (bit 11) 64 is set, a shift and mask are executed on the Bus Received Register 20 operand through MUX 32, and then a merge of the contents of the Bus Received Register 20 and**

> **Register A 22 from MUX 30 is executed, according to the control bits specified in the Mask and Merge operand stored in the Mask/Merge Register 24. When the R field 64 is clear, a shift and mask are executed on the contents of Register A 22, and then a merge of Register A 22 and the Bus Received Register 20 is executed, according to the control bits specified in the Mask and Merge operand stored in the Mask/Merge Register 24. After these operations are performed, the result is in Accumulator 26. The result is then stored at the address in the Local Store 16 memory location designated by the Destination field (bits 33-43) 54. (emphasis added, col. 5, line 6, to col. 6, line 15)**

Thus, Byers' mask, which is used for performing the mask and merge operations, is loaded from the local store 16 into a specialized Mask/Merge Register 24. Therefore, Byers' instruction does not specify the actual mask as recited in applicant's claims, but rather the location in the control store where the mask value that controls the mask/merge operations is stored. Indeed, Byers explains that "[m]ask and merge indicator bits control bit pairs in the preferred embodiment of this invention rather than individual bits of the operands. If an indicator bit was to be used for each bit in the operand to be masked or merged, a 36-bit register would be needed for each of the Mask and Merge indicator sets. Thus, an additional register would need to be allocated in this architecture to support masking and merging in a single instruction." Thus, even with the economized mask/merge configuration employed by Byers where each bit of the mask corresponds to a pair of bits of the operand, the Byers' mask is too long to be specified within Byers' instruction. Accordingly, Byers does not disclose or suggest at least the feature of "with loading using an associated bit mask specified in the instruction with each bit of the associated bit mask identifying a different byte of the transfer register," as required by applicant's independent claim 1.

Hennessey's "Computer Organization and Design" generally describes the benefits of connecting multiple processors together (see page 712). However, Hennessy does not provide any technical details regarding the configuration and operation of such interconnected multiple processors. Hennessey's "Computer organization and Design" also does not disclose or suggest instructions or operations that cause a processor to selectively load any specified combination of bytes of data within a transfer register, and certainly does not describe use of a mask specified within instructions to facilitate selectively loading data Therefore, Hennessey's "Computer organization and Design" also does not disclose or suggest at least the feature of "with loading using an associated bit mask specified in the instruction with each bit of the associated bit mask identifying a different byte of the transfer register," as required by applicant's independent claim 1.

Hennessy's "Computer Architecture A Quantitative Approach" neither describes nor suggests instructions or operations that cause the ALU to selectively load any specified combination of bytes of data within a register nor instructions that include a mask, specified in the instruction itself, to selectively load data. Thus, Hennessy's "Computer Architecture A Quantitative Approach" does not disclose or suggest at least the feature of "with loading using an associated bit mask specified in the instruction with each bit of the associated bit mask identifying a different byte of the transfer register," as required by applicant's independent claim 1.

Since none of the cited references discloses or suggests, alone or in combination, at least the feature of "with loading using an associated bit mask specified in the instruction with each bit of the associated bit mask identifying a different byte of the transfer register," independent claim 1 is therefore patentable over the cited references. Claims 3-9 depend from claim 1 and are therefore patentable for at least the same reasons as independent claim 1.

Independent claims 10 and 19 recite the feature of "selectively loading any combination of bytes of data within a register associated with one of a plurality of microengines with a shifted value of an operand, using an associated bit mask specified in an instruction, with each bit of the associated bit mask identifying a different byte of the register," or similar language. For reasons similar to those provided with respect to independent claim 1, at least this feature is not disclosed by the cited art. Accordingly, independent claims 10 and 19 are patentable over the cited art. Claims 12-18 depend from independent claim 10 and are therefore patentable for at least the same reasons as claim 10. Claims 21-27 depend from independent claim 19 and are therefore patentable for at least the same reasons as independent claim 19.

In view of the foregoing remarks, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

Canceled claims, if any, have been canceled without prejudice or disclaimer. Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

No fees are believed due. Please apply any other charges to deposit account 06-1050, referencing attorney docket 10559-309US1.

Respectfully submitted,

Date:May 24, 2006

/Ido Rabinovitch/
Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

Fish & Richardson P.C.
P. O. Box 1022
Minneapolis, MN 55440-1022
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21338520.doc